Reformulações e Algoritmos de Solução para o Problema de Árvore Geradora com Máximo número de Folhas

Luidi G. Simonetti
Abilio Lucena
Nelson Maculan

IC - UNICAMP

24/10/2008

Outline of Presentation

Background Material

Max Leaf Spanning Tree Problem

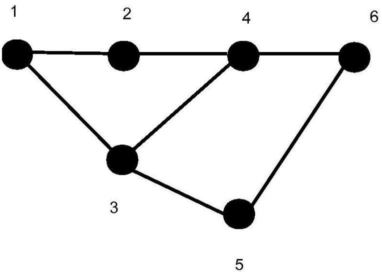
Formulation + Reformulations

Computational Results

Conclusions

Graph G

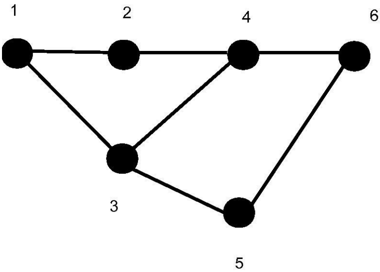
Graph $G = (V, E)$: vertex set V and edge set E.

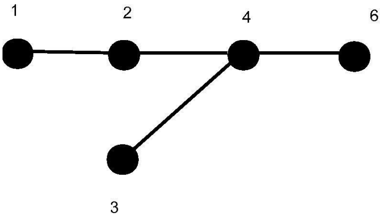


Tree $\mathcal{T}$ of G

$\mathcal{T} = (V', E')$, $V' \subseteq V$, $E' \subseteq E$

- Connected subgraph of G with $|E'| = |V'| - 1$
- Not necessarily spanning for G
- $\mathcal{T}$ may eventually consist of a single vertex of G





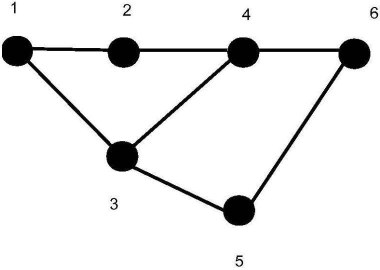
Leaf of $\mathcal{T}$ vertex of $\mathcal{T}$ incident to only one other vertex in $\mathcal{T}$

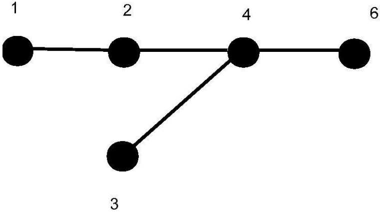
UNICAMP

Tree $\mathcal{T}$ of G

$\mathcal{T} = (V', E')$, $V' \subseteq V$, $E' \subseteq E$

- Connected subgraph of G with $|E'| = |V'| - 1$
- Not necessarily spanning for G
- $\mathcal{T}$ may eventually consist of a single vertex of G





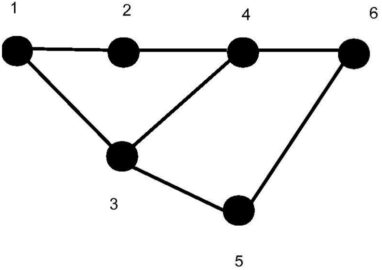
Leaf of $\mathcal{T}$ vertex of $\mathcal{T}$ incident to only one other vertex in $\mathcal{T}$

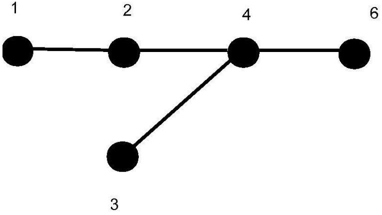
UNICAMP

Tree $\mathcal{T}$ of G

$\mathcal{T} = (V', E')$, $V' \subseteq V$, $E' \subseteq E$

- Connected subgraph of G with $|E'| = |V'| - 1$
- Not necessarily spanning for G
- $\mathcal{T}$ may eventually consist of a single vertex of G



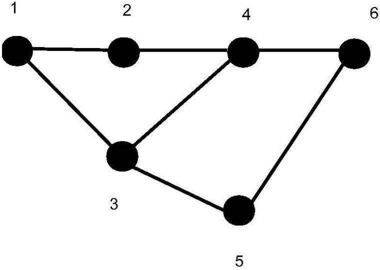


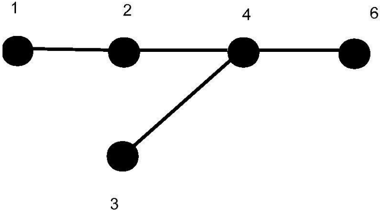
Leaf of $\mathcal{T}$ vertex of $\mathcal{T}$ incident to only one other vertex in $\mathcal{T}$

Tree $\mathcal{T}$ of G

$\mathcal{T} = (V', E')$, $V' \subseteq V$, $E' \subseteq E$

- Connected subgraph of G with $|E'| = |V'| - 1$
- Not necessarily spanning for G
- $\mathcal{T}$ may eventually consist of a single vertex of G

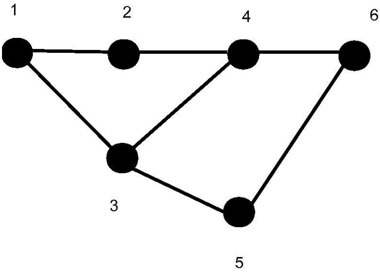


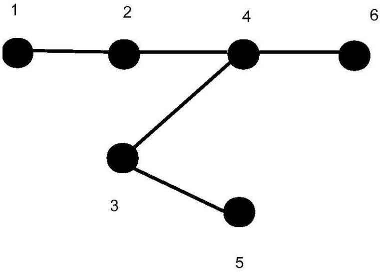


Leaf of $\mathcal{T}$ vertex of $\mathcal{T}$ incident to only one other vertex in $\mathcal{T}$

Spanning trees of G

$\mathcal{T} = (V', E')$, $V' = V$, $E' \subseteq E$: All vertices of V are involved!





Minimum Spanning Tree Problem (MSTP)

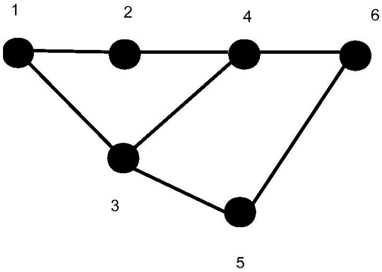
Given: $G = (V, E)$ and $\{c_e : e \in E\}$

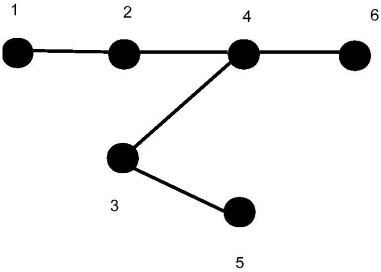
Find: least cost spanning tree of G

Polynomial time solution algorithm (Kruskal (1956))

Spanning trees of G

$\mathcal{T} = (V', E')$, $V' = V$, $E' \subseteq E$: All vertices of V are involved!





Minimum Spanning Tree Problem (MSTP)

Given: $G = (V, E)$ and $\{c_e : e \in E\}$

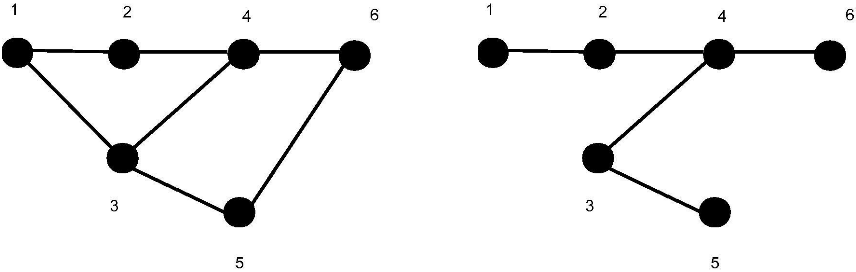
Find: least cost spanning tree of G

Polynomial time solution algorithm (Kruskal [1958])

UNICAMP

Spanning trees of G

$\mathcal{T} = (V', E')$, $V' = V$, $E' \subseteq E$: All vertices of V are involved!



Minimum Spanning Tree Problem (MSTP)

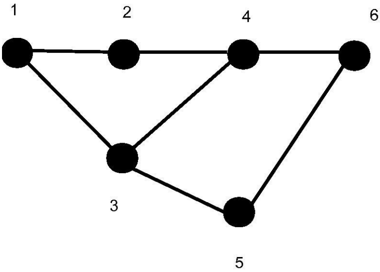
Given: $G = (V, E)$ and $\{c_e : e \in E\}$

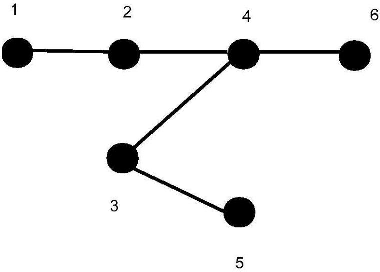
Find: least cost spanning tree of G

Polynomial time solution algorithm (Kruskal [1958])

Spanning trees of G

$\mathcal{T} = (V', E')$, $V' = V$, $E' \subseteq E$: All vertices of V are involved!





Minimum Spanning Tree Problem (MSTP)

Given: $G = (V, E)$ and $\{c_e : e \in E\}$

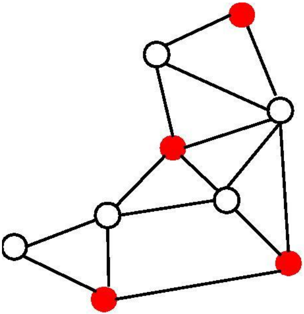
Find: least cost spanning tree of G

Polynomial time solution algorithm (Kruskal [1958])

Steiner Tree Problem in Graphs (STPG)

Given: $G = (V, E)$, $T \subseteq V$ and $\{c_e : e \in E\}$

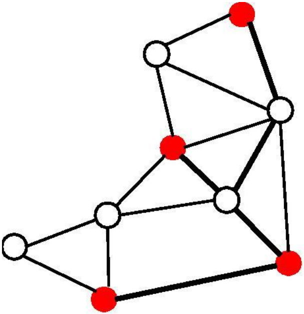
Find: least cost tree of G that spans T



Steiner Tree Problem in Graphs (STPG)

Given: $G = (V, E)$, $T \subseteq V$ and $\{c_e : e \in E\}$

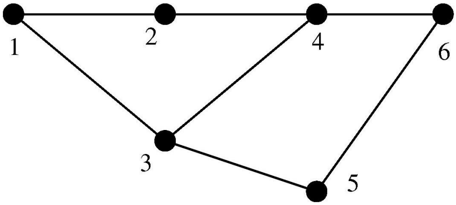
Find: least cost tree of G that spans T



Max Leaf Spanning Tree Problem (MLSTP)

Given: a graph $G = (V, E)$

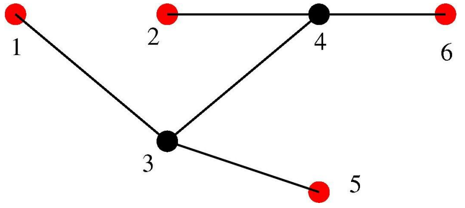
Find: spanning tree of G with the largest number of leaves



Max Leaf Spanning Tree Problem (MLSTP)

Given: a graph $G = (V, E)$

Find: spanning tree of G with the largest number of leaves

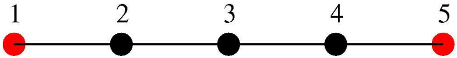


Optimal solution has 4 leaves.

Max Leaf Spanning Tree Problem (MLSTP)

Given: a graph $G = (V, E)$

Find: spanning tree of G with the largest number of leaves

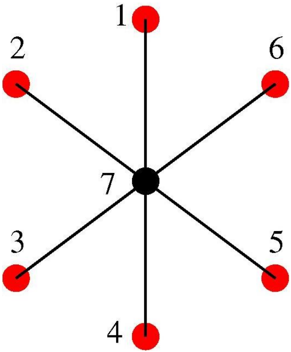


Worst case - Difficult to find optimal solution: 2 leaves.

Max Leaf Spanning Tree Problem (MLSTP)

Given: a graph $G = (V, E)$

Find: spanning tree of G with the largest number of leaves



Best case - Trivial to obtain optimal solution
for complete graphs: $(|V| - 1)$ leaves

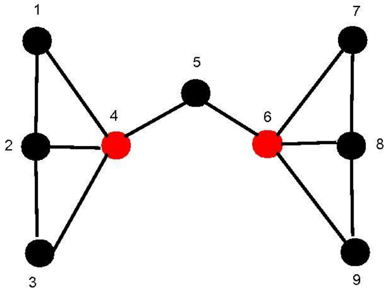
Equivalent to the problem

Dominating set of G: $C \subset V$ such that

- for every $i \in V \setminus C$: edge with an endpoint in i and the other in C.

Minimum Connected Dominating set C (MCDS)

- Implies a tree of G for non-leaf vertices of $\mathcal{T}$.
- Straightforward to generate a Max Leaf Spanning Tree of G.



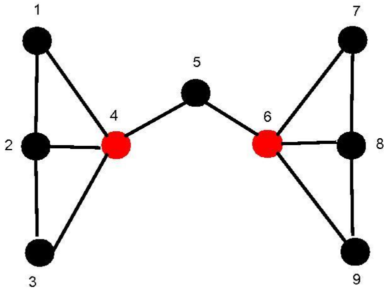
Equivalent to the problem

Minimun Dominating set of G: $C \subset V$ such that

- for every $i \in V \setminus C$: edge with an endpoint in i and the other in C.
- find a dominating set of G with as few vertices as possible.

Minimum Connected Dominating set C (MCDS)

- Implies a tree of G for non-leaf vertices of $\mathcal{T}$.
- Straightforward to generate a Max Leaf Spanning Tree of G.



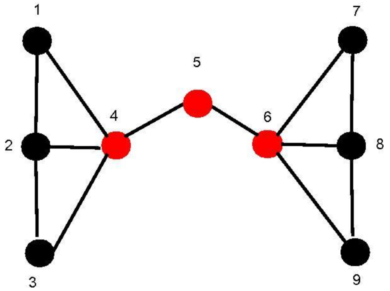
Equivalent to the problem

Minimun Dominating set of G: $C \subset V$ such that

- for every $i \in V \setminus C$: edge with an endpoint in i and the other in C.
- find a dominating set of G with as few vertices as possible.

Minimum Connected Dominating set C (MCDS)

- Implies a tree of G for non-leaf vertices of $\mathcal{T}$.
- Straightforward to generate a Max Leaf Spanning Tree of G.



Why studying MLSTP?

- *NP*-hard
- Model for some telecommunications networks
- Models some circuit layout problems

Why studying MLSTP?

- *NP*-hard
- Model for some telecommunications networks
- Models some circuit layout problems

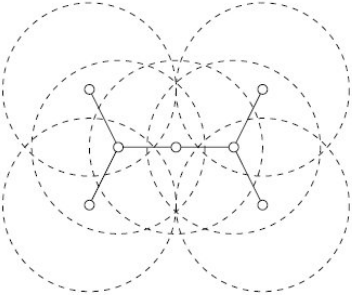
UNICAMP

Why studying MLSTP?

- *NP*-hard
- Model for some telecommunications networks
- Models some circuit layout problems

Why studying MLSTP?

- *NP*-hard
- Model for some telecommunications networks
  Ex: Wireless Ad Hoc Networks ⇒ virtual backbone ⇒ MCDS
- Models some circuit layout problems



Previous Work

Garey and Johnson (1979): problem is *NP*-hard.

Lu and Ravi (1992,1998): factor of 3 approximation algorithm.

Galbiati, Maffioli and Morzenti (1994): problem is *SNP*-hard.
There exists ϵ such that finding $(1+\epsilon)$-approximation algorithm is *NP*-hard

Solis-Oba (1998): factor of 2 approximation algorithm.

Fernandes and Gouveia (1998): formulation for fixed number of leaves.

Fujie (2003): formulation and Branch and Bound algorithm.

Fujie (2004): new formulation and polyhedral study.
No computational results

Previous Work

Garey and Johnson (1979): problem is *NP*-hard.

Lu and Ravi (1992,1998): factor of 3 approximation algorithm.

Galbiati, Maffioli and Morzenti (1994): problem is *SNP*-hard.
There exists ϵ such that finding $(1+\epsilon)$-approximation algorithm is *NP*-hard

Solis-Oba (1998): factor of 2 approximation algorithm.

Fernandes and Gouveia (1998): formulation for fixed number of leaves.

Fujie (2003): formulation and Branch and Bound algorithm.

Fujie (2004): new formulation and polyhedral study.
No computational results

Previous Work

Garey and Johnson (1979): problem is *NP*-hard.

Lu and Ravi (1992,1998): factor of 3 approximation algorithm.

Galbiati, Maffioli and Morzenti (1994): problem is *SNP*-hard.
There exists ϵ such that finding $(1 + \epsilon)$-approximation algorithm is *NP*-hard

Solis-Oba (1998): factor of 2 approximation algorithm.

Fernandes and Gouveia (1998): formulation for fixed number of leaves.

Fujie (2003): formulation and Branch and Bound algorithm.

Fujie (2004): new formulation and polyhedral study.
No computational results

UNICAMP

Previous Work

Garey and Johnson (1979): problem is *NP*-hard.

Lu and Ravi (1992,1998): factor of 3 approximation algorithm.

Galbiati, Maffioli and Morzenti (1994): problem is *SNP*-hard.
There exists ϵ such that finding $(1 + \epsilon)$-approximation algorithm is *NP*-hard

Solis-Oba (1998): factor of 2 approximation algorithm.

Fernandes and Gouveia (1998): formulation for fixed number of leaves.

Fujie (2003): formulation and Branch and Bound algorithm.

Fujie (2004): new formulation and polyhedral study.
No computational results

UNICAMP

Previous Work

Garey and Johnson (1979): problem is *NP*-hard.

Lu and Ravi (1992,1998): factor of 3 approximation algorithm.

Galbiati, Maffioli and Morzenti (1994): problem is *SNP*-hard.
There exists ϵ such that finding $(1+\epsilon)$-approximation algorithm is *NP*-hard

Solis-Oba (1998): factor of 2 approximation algorithm.

Fernandes and Gouveia (1998): formulation for fixed number of leaves.

Fujie (2003): formulation and Branch and Bound algorithm.

Fujie (2004): new formulation and polyhedral study.
No computational results

UNICAMP

Previous Work

Garey and Johnson (1979): problem is *NP*-hard.

Lu and Ravi (1992,1998): factor of 3 approximation algorithm.

Galbiati, Maffioli and Morzenti (1994): problem is *SNP*-hard.
There exists ϵ such that finding $(1+\epsilon)$-approximation algorithm is *NP*-hard

Solis-Oba (1998): factor of 2 approximation algorithm.

Fernandes and Gouveia (1998): formulation for fixed number of leaves.

Fujie (2003): formulation and Branch and Bound algorithm.

Fujie (2004): new formulation and polyhedral study.
No computational results

Previous Work

Garey and Johnson (1979): problem is *NP*-hard.

Lu and Ravi (1992,1998): factor of 3 approximation algorithm.

Galbiati, Maffioli and Morzenti (1994): problem is *SNP*-hard.
There exists ϵ such that finding $(1+\epsilon)$-approximation algorithm is *NP*-hard

Solis-Oba (1998): factor of 2 approximation algorithm.

Fernandes and Gouveia (1998): formulation for fixed number of leaves.

Fujie (2003): formulation and Branch and Bound algorithm.

Fujie (2004): new formulation and polyhedral study.
No computational results

Formulation of Fujie [2003,2004]

Variables involved:

- $x = \{x_e \in \mathbb{R}_+ : e \in E\}$: for the edges of G.
- $z = \{z_i \in \{0,1\} : i \in V\}$: to represent leaves of $\mathcal{T}$.

Notation:

- $\delta(i) \subseteq E$, $i \in V$: edges incident to i.
- $E(S) \subseteq E$, $S \subseteq V$: edges with both endpoints in S.

$$
\begin{aligned}
\max \sum_{i \in V} z_i & & \\
\text{s.t.} \sum_{e \in E} x_e &= |V| - 1 & \\
\sum_{e \in E(S)} x_e &\leq |S| - 1 & S \subset V, |S| \geq 2 \\
\sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i &\leq |\delta(i)| & i \in V \\
0 \leq x_e &\leq 1 & e \in E \\
z_i &\in \{0,1\} & i \in V.
\end{aligned}
$$

Formulation of Fujie [2003,2004]

Variables involved:

- $x = \{x_e \in \mathbb{R}_+ : e \in E\}$: for the edges of G.
- $z = \{z_i \in \{0,1\} : i \in V\}$: to represent leaves of $\mathcal{T}$.

Notation:

- $\delta(i) \subseteq E$, $i \in V$: edges incident to i.
- $E(S) \subseteq E$, $S \subseteq V$: edges with both endpoints in S.

$$
\begin{aligned}
\max \sum_{i \in V} z_i & \\
\text{s.t.} \sum_{e \in E} x_e = |V| - 1 & \\
\sum_{e \in E(S)} x_e \leq |S| - 1 & \qquad S \subset V, |S| \geq 2 \\
\sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i \leq |\delta(i)| & \qquad i \in V \\
0 \leq x_e \leq 1 & \qquad e \in E \\
z_i \in \{0,1\} & \qquad i \in V.
\end{aligned}
$$

Formulation of Fujie [2003,2004]

Variables involved:

- $x = \{x_e \in \mathbb{R}_+ : e \in E\}$: for the edges of G.
- $z = \{z_i \in \{0,1\} : i \in V\}$: to represent leaves of $\mathcal{T}$.

Notation:

- $\delta(i) \subseteq E$, $i \in V$: edges incident to i.
- $E(S) \subseteq E$, $S \subseteq V$: edges with both endpoints in S.

$$
\begin{aligned}
& \max \sum_{i \in V} z_i \\
\text{s.t.} & \sum_{e \in E} x_e = |V| - 1 \\
& \sum_{e \in E(S)} x_e \leq |S| - 1 && S \subset V, |S| \geq 2 \\
& \sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i \leq |\delta(i)| && i \in V \\
& 0 \leq x_e \leq 1 && e \in E \\
& z_i \in \{0,1\} && i \in V.
\end{aligned}
$$

Formulation of Fujie [2003,2004]

Variables involved:

- $x = \{x_e \in \mathbb{R}_+ : e \in E\}$: for the edges of G.
- $z = \{z_i \in \{0,1\} : i \in V\}$: to represent leaves of $\mathcal{T}$.

Notation:

- $\delta(i) \subseteq E$, $i \in V$: edges incident to i.
- $E(S) \subseteq E$, $S \subseteq V$: edges with both endpoints in S.

$$
\begin{aligned}
\max \sum_{i \in V} z_i & \\
\text{s.t.} \sum_{e \in E} x_e = |V| - 1 & \\
\sum_{e \in E(S)} x_e \leq |S| - 1 & \qquad S \subset V, |S| \geq 2 \\
\sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i \leq |\delta(i)| & \qquad i \in V \\
0 \leq x_e \leq 1 & \qquad e \in E \\
z_i \in \{0,1\} & \qquad i \in V.
\end{aligned}
$$

UNICAMP

Formulation of Fujie [2003,2004]

Variables involved:

- $x = \{x_e \in \mathbb{R}_+ : e \in E\}$: for the edges of G.
- $z = \{z_i \in \{0,1\} : i \in V\}$: to represent leaves of $\mathcal{T}$.

Notation:

- $\delta(i) \subseteq E$, $i \in V$: edges incident to i.
- $E(S) \subseteq E$, $S \subseteq V$: edges with both endpoints in S.

$$
\begin{aligned}
\max \sum_{i \in V} z_i & \\
\text{s.t.} \sum_{e \in E} x_e = |V| - 1 & \\
\sum_{e \in E(S)} x_e \leq |S| - 1 & \qquad S \subset V, |S| \geq 2 \\
\sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i \leq |\delta(i)| & \qquad i \in V \\
0 \leq x_e \leq 1 & \qquad e \in E \\
z_i \in \{0,1\} & \qquad i \in V.
\end{aligned}
$$

UNICAMP

Facet Defining Inequalities

Recall that inequality

$$\sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i \leq |\delta(i)| \qquad i \in V$$

is used in formulation.

Then, under some mild assumptions - Fujie[2004],

$$\sum_{e \in F} x_e + (|F| - 1) z_i \leq |F| \qquad i \in V, F \subseteq \delta(i), |F| \geq 2$$

are facet defining inequalities for the polytope defined by the formulation.

Relaxation of Fujie's formulation

Formulation may be expressed as:

$$\max \sum_{i \in V} z_i$$

$$\text{s.t.} \quad \sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i \leq |\delta(i)| \qquad i \in V$$

$$x \in ST_G$$

$$z_i \in \{0, 1\} \qquad i \in V$$

ST_G: set of all incidence vectors of spanning trees in G.

Relaxation:

- relax integrality constraints on z to $0 \leq z \leq 1$.

Relaxation of Fujie's formulation

Formulation may be expressed as:

$$\begin{aligned} & \max \sum_{i \in V} z_i & \\ \text{s.t.} \quad & \sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i \le |\delta(i)| & i \in V \\ & x \in ST_G & \\ & z_i \in \{0, 1\} & i \in V \end{aligned}$$

ST_G: set of all incidence vectors of spanning trees in G.

Relaxation:

- relax integrality constraints on z to $0 \le z \le 1$.

UNICAMP

Relaxation of Fujie's formulation (continuation)

Since one is maximizing $\sum_{i \in V} z_i$:

$$\sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i \leq |\delta(i)|, i \in V,$$

must hold as equality at an optimal solution!

One may then write:

$$\begin{aligned} \max \quad & \sum_{i \in V} \frac{|\delta(i)| - \sum_{e \in \delta(i)} x_e}{|\delta(i)| - 1} \\ = \quad & \sum_{i \in V} \frac{|\delta(i)|}{|\delta(i)| - 1} - \sum_{e=(i,j) \in E} \left(\frac{1}{|\delta(i)| - 1} + \frac{1}{|\delta(j)| - 1} \right) x_e \\ \text{s.t.} \quad & x \in ST_G. \end{aligned}$$

- Relaxation is easy to solve: MSTP
- Relaxation bound equals LP-relaxation bound for original formulation.
- Branch-and-Bound algorithm based on solving MSTPs!

UNICAMP

Relaxation of Fujie's formulation (continuation)

Since one is maximizing $\sum_{i \in V} z_i$:

$$\sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i \leq |\delta(i)|, i \in V,$$

must hold as equality at an optimal solution!

One may then write:

$$\begin{aligned} \max \quad & \sum_{i \in V} \frac{|\delta(i)| - \sum_{e \in \delta(i)} x_e}{|\delta(i)| - 1} \\ = \quad & \sum_{i \in V} \frac{|\delta(i)|}{|\delta(i)| - 1} - \sum_{e=(i,j) \in E} \left(\frac{1}{|\delta(i)| - 1} + \frac{1}{|\delta(j)| - 1} \right) x_e \\ \text{s.t.} \quad & x \in ST_G. \end{aligned}$$

- Relaxation is easy to solve: MSTP
- Relaxation bound equals LP-relaxation bound for original formulation.
- Branch-and-Bound algorithm based on solving MSTPs!

UNICAMP

Relaxation of Fujie's formulation (continuation)

Since one is maximizing $\sum_{i \in V} z_i$:

$$\sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i \leq |\delta(i)|, i \in V,$$

must hold as equality at an optimal solution!

One may then write:

$$\begin{aligned} \max \quad & \sum_{i \in V} \frac{|\delta(i)| - \sum_{e \in \delta(i)} x_e}{|\delta(i)| - 1} \\ = \quad & \sum_{i \in V} \frac{|\delta(i)|}{|\delta(i)| - 1} - \sum_{e=(i,j) \in E} \left(\frac{1}{|\delta(i)| - 1} + \frac{1}{|\delta(j)| - 1} \right) x_e \\ \text{s.t.} \quad & x \in ST_G. \end{aligned}$$

- Relaxation is easy to solve: MSTP
- Relaxation bound equals LP-relaxation bound for original formulation.
- Branch-and-Bound algorithm based on solving MSTPs!

UNICAMP

Relaxation of Fujie's formulation (continuation)

Since one is maximizing $\sum_{i \in V} z_i$:

$$\sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i \leq |\delta(i)|, i \in V,$$

must hold as equality at an optimal solution!

One may then write:

$$\begin{aligned} \max \quad & \sum_{i \in V} \frac{|\delta(i)| - \sum_{e \in \delta(i)} x_e}{|\delta(i)| - 1} \\ = \quad & \sum_{i \in V} \frac{|\delta(i)|}{|\delta(i)| - 1} - \sum_{e=(i,j) \in E} \left(\frac{1}{|\delta(i)| - 1} + \frac{1}{|\delta(j)| - 1} \right) x_e \\ \text{s.t.} \quad & x \in ST_G. \end{aligned}$$

- Relaxation is easy to solve: MSTP
- Relaxation bound equals LP-relaxation bound for original formulation.
- Branch-and-Bound algorithm based on solving MSTPs!

UNICAMP

Relaxation of Fujie's formulation (continuation)

Since one is maximizing $\sum_{i \in V} z_i$:

$$\sum_{e \in \delta(i)} x_e + (|\delta(i)| - 1) z_i \leq |\delta(i)|, i \in V,$$

must hold as equality at an optimal solution!

One may then write:

$$\begin{aligned} \max \quad & \sum_{i \in V} \frac{|\delta(i)| - \sum_{e \in \delta(i)} x_e}{|\delta(i)| - 1} \\ = \quad & \sum_{i \in V} \frac{|\delta(i)|}{|\delta(i)| - 1} - \sum_{e=(i,j) \in E} \left(\frac{1}{|\delta(i)| - 1} + \frac{1}{|\delta(j)| - 1} \right) x_e \\ \text{s.t.} \quad & x \in ST_G. \end{aligned}$$

- Relaxation is easy to solve: MSTP
- Relaxation bound equals LP-relaxation bound for original formulation.
- Branch-and-Bound algorithm based on solving MSTPs!

Reformulation Strategies for MLSTP

Strategy 1: Directed graph reformulation
Max Leaf Spanning Arborescence Problem.

Strategy 2: Steiner Arborescence Problem on a Level Graph.

First MLSTP Reformulation

- Directed graph reformulation of Fujie's formulation: $x_e = y_{ij} + y_{ji}$, $e = [i,j] \in E$.
- $\delta^+(i)$ (resp. $\delta^-(i)$): arcs leaving (resp. entering) $i \in V$

$$
\begin{aligned}
\max \sum_{i \in V} z_i & \\
\text{s.t.} \sum_{a \in A} y_a = |V| - 1 & \\
\sum_{a \in \delta^-(j)} y_a = 1 & \qquad j \in V \setminus \{r\} \\
\sum_{a \in A(S)} y_a \leq |S| - 1 & \qquad S \subset V, |S| \geq 2 \\
\sum_{a \in \delta^-(i)} y_a + \sum_{a \in \delta^+(i)} y_a + (|\delta^+(i)| - 1) z_i \leq |\delta^+(i)| & \qquad i \in V \setminus \{r\} \\
\sum_{a \in \delta^+(r)} y_a + (|\delta^+(r)| - 1) z_r \leq |\delta^+(r)| & \\
0 \leq y_a \leq 1 & \qquad a \in A \\
z_i \in \{0, 1\} & \qquad i \in V
\end{aligned}
$$

UNICAMP

Advantage over Fujie's formulation

- Valid inequalities: $y_a + z_i \leq 1, \quad a = (i,j) \in A$
  no equivalent inequality in undirected formulation
- Since $x_e = y_{ij} + y_{ji}$, for $e = [i,j] \in E$, Fujie's valid inequalities are re-written as:

$$\sum_{a \in F} y_a + \sum_{a=(j,i) \in A | (i,j) \in F} y_a + (|F| - 1)z_i \leq |F|, \; i \in V, \quad F \subseteq \delta^+(i), \quad |F| \geq 2.$$

- Branch-and-Cut algorithm based on reinforced formulation!

UNICAMP

Advantage over Fujie's formulation

- Valid inequalities: $y_a + z_i \leq 1, \quad a = (i, j) \in A$
  no equivalent inequality in undirected formulation
- Since $x_e = y_{ij} + y_{ji}$, for $e = [i, j] \in E$, Fujie's valid inequalities are re-written as:

$$\sum_{a \in F} y_a + \sum_{a=(j,i) \in A | (i,j) \in F} y_a + (|F| - 1) z_i \leq |F|, \; i \in V, \quad F \subseteq \delta^+(i), \quad |F| \geq 2.$$

- Branch-and-Cut algorithm based on reinforced formulation!

UNICAMP

Advantage over Fujie's formulation

- Valid inequalities: $y_a + z_i \leq 1, \quad a = (i,j) \in A$
  no equivalent inequality in undirected formulation
- Since $x_e = y_{ij} + y_{ji}$, for $e = [i,j] \in E$, Fujie's valid inequalities are re-written as:

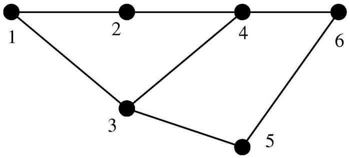
$$\sum_{a \in F} y_a + \sum_{a=(j,i) \in A | (i,j) \in F} y_a + (|F| - 1) z_i \leq |F|, \; i \in V, \quad F \subseteq \delta^+(i), \quad |F| \geq 2.$$

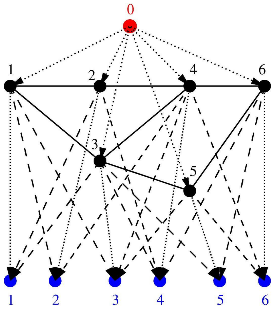
- Branch-and-Cut algorithm based on reinforced formulation!

Second Reformulation for MLSTP

Steiner Arborescence Problem based on Level Graphs $D_F = (V_F, A_F)$

$$V_F = \{0\} \cup \{(i,h) : 1 \leq h \leq 2,\ i \in V\}$$
$$A_F = \{(0,(j,1)) : j \in V\} \cup \{((i,1),(j,1)) : [i,j] \in A\}$$
$$\cup \{((i,1),(i,2)) : i \in V\} \cup \{((i,1),(j,2)) : [i,j] \in A\}$$

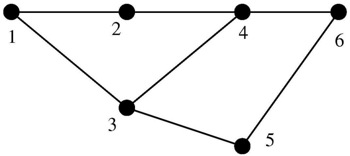


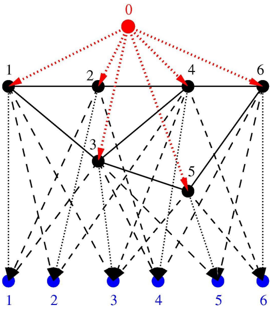


Second Reformulation for MLSTP

Steiner Arborescence Problem based on Level Graphs $D_F = (V_F, A_F)$

$$V_F = \{0\} \cup \{(i,h) : 1 \le h \le 2,\ i \in V\}$$
$$A_F = \{(0,(j,1)) : j \in V\} \cup \{((i,1),(j,1)) : [i,j] \in A\}$$
$$\cup \{((i,1),(i,2)) : i \in V\} \cup \{((i,1),(j,2)) : [i,j] \in A\}$$



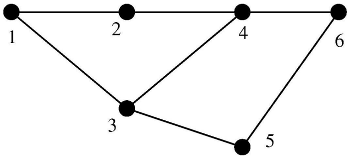


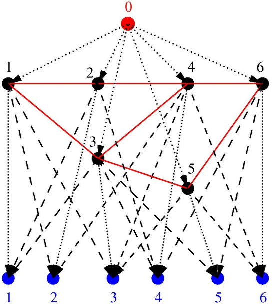
$c_{0j} \rightarrow \infty$ - only one arc $(0,(j,1))$

Second Reformulation for MLSTP

Steiner Arborescence Problem based on Level Graphs $D_F = (V_F, A_F)$

$$V_F = \{0\} \cup \{(i,h) : 1 \le h \le 2,\ i \in V\}$$
$$A_F = \{(0,(j,1)) : j \in V\} \cup \{((i,1),(j,1)) : [i,j] \in A\}$$
$$\cup\ \{((i,1),(i,2)) : i \in V\} \cup \{((i,1),(j,2)) : [i,j] \in A\}$$



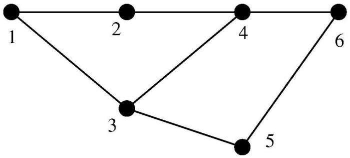


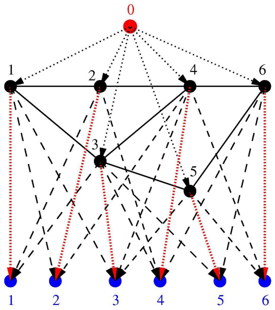
$c_{ij}^1 = |V| - 1$

Second Reformulation for MLSTP

Steiner Arborescence Problem based on Level Graphs $D_F = (V_F, A_F)$

$$
\begin{aligned}
V_F &= \{0\} \cup \{(i,h) : 1 \leq h \leq 2,\ i \in V\} \\
A_F &= \{(0,(j,1)) : j \in V\} \cup \ \{((i,1),(j,1)) : [i,j] \in A\} \\
&\quad \cup \ \{((i,1),(i,2)) : i \in V\} \cup \ \{((i,1),(j,2)) : [i,j] \in A\}
\end{aligned}
$$



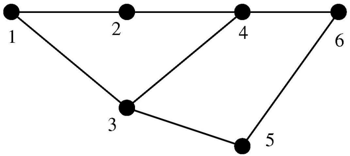


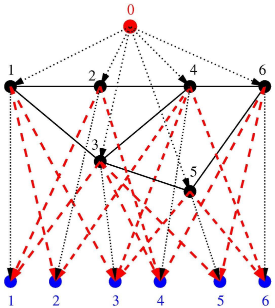
$c_{ii} = 0$

Second Reformulation for MLSTP

Steiner Arborescence Problem based on Level Graphs $D_F = (V_F, A_F)$

$$
\begin{aligned}
V_F &= \{0\} \cup \{(i,h) : 1 \leq h \leq 2,\ i \in V\} \\
A_F &= \{(0,(j,1)) : j \in V\} \cup \{((i,1),(j,1)) : [i,j] \in A\} \\
&\quad \cup \{((i,1),(i,2)) : i \in V\} \cup \{((i,1),(j,2)) : [i,j] \in A\}
\end{aligned}
$$

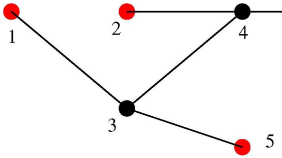


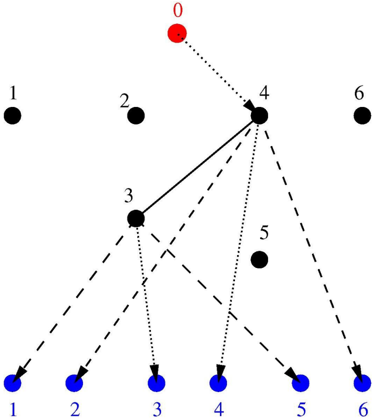


$c_{ij}^2 = 1$

Steiner Arborescence Reformulation for MLSTP

- Set of terminal vertices: $T = \{(i, 2) : \ i \in V\}$.
- Optimal solution for MLSTP in $G \Rightarrow$ Steiner tree in G_F, rooted at 0.
- Under conveniently defined costs, SAP reformulation for *MLSTP*.





Computational Results

|V|	dens.	opt.	B&B			Dir			SAP		
			LP bound	# nodes	T(s)	LP bound	# nodes	T(s)	LP bound	# nodes	T(s)
	10	15	17.83	291	0.12	15.57	1	0.01	15.80	1	0.04
	20	23	26.04	5055	0.33	24.48	7	0.1	23.95	1	0.12
30	30	26	27.44	842	0.24	27.05	1	0.03	26.13	5	26.7
	50	27	28.46	307	0.19	28.13	3	0.09	27.94	1	1.28
	70	28	28.83	1	0.16	28.73	1	0.01	28.00	1	0.26
	5	19	21.50	265	0.94	19.00	1	0.02	19.00	1	0.09
	10	38	42.16	82599	4.54	39.75	41	0.82	38.86	38	94
50	20	43	46.58	225771	16.9	45.22	77	1.32	44.48	57	1827
	30	45	47.59	38155	5.97	46.80	39	1.21	46.08	43	22424
	50	47	48.45	3050	4	48.18	13	0.51	47.36	-	-
	70	48	48.82	5	1.64	48.62	1	0.09	48.00	1	2.08
	5	43	49.57	9068999	313	44.45	53	0.99	43.56	19	103
	10	57	63.18	-	-	59.20	174	4.73	58.60	-	-
70	20	63	66.22	-	-	65.04	607	16.3	64.37	-	-
	30	65	67.62	4113677	536	66.91	35	2.9	66.15		-
	50	67	68.52	33058	25.3	68.14	7	1.33	-	-	-
	70	68	68.76	2661	10.8	68.68	5	1.92	68	1	8.55

Computational Results

$\|V\|$	dens.	opt.	B&B			Dir			SAP		
			LP bound	# nodes	T(s)	LP bound	# nodes	T(s)	LP bound	# nodes	T(s)
	10	15	17.83	291	0.12	15.57	1	0.01	15.80	1	0.04
	20	23	26.04	5055	0.33	24.48	7	0.1	23.95	1	0.12
30	30	26	27.44	842	0.24	27.05	1	0.03	26.13	5	26.7
	50	27	28.46	307	0.19	28.13	3	0.09	27.94	1	1.28
	70	28	28.83	1	0.16	28.73	1	0.01	28.00	1	0.26
	5	19	21.50	265	0.94	19.00	1	0.02	19.00	1	0.09
	10	38	42.16	82599	4.54	39.75	41	0.82	38.86	38	94
50	20	43	46.58	225771	16.9	45.22	77	1.32	44.48	57	1827
	30	45	47.59	38155	5.97	46.80	39	1.21	46.08	43	22424
	50	47	48.45	3050	4	48.18	13	0.51	47.36	-	-
	70	48	48.82	5	1.64	48.62	1	0.09	48.00	1	2.08
	5	43	49.57	9068999	313	44.45	53	0.99	43.56	19	103
	10	57	63.18	-	-	59.20	174	4.73	58.60	-	-
70	20	63	66.22	-	-	65.04	607	16.3	64.37	-	-
	30	65	67.62	4113677	536	66.91	35	2.9	66.15		-
	50	67	68.52	33058	25.3	68.14	7	1.33	-	-	-
	70	68	68.76	2661	10.8	68.68	5	1.92	68	1	8.55

Computational Results

|V|	dens.	opt.	B&B			Dir			SAP		
			LP bound	# nodes	T(s)	LP bound	# nodes	T(s)	LP bound	# nodes	T(s)
100	5	76	83.42	-	-	79.36	605	24.5	78.57	-	-
	10	87	93.27	-	-	89.52	135	9.36	-	-	-
	20	92	96.56	-	-	94.85	1025	86.1	-	-	-
	30	94	97.39	-	-	96.68	1753	258	-	-	-
	50	96	98.36	348389	213	98.03	479	132	-	-	-
	70	97	98.76	9091	50.5	98.64	121	154	-	-	-
120	5	95	105.04	-	-	97.77	24	2.65	-	-	-
	10	107	113.16	-	-	109.83	869	65.4	-	-	-
	20	112	116.39	-	-	114.93	2401	393	-	-	-
	30	114	117.40	-	-	116.69	2301	653	-	-	-
	50	116	118.42	571335	435	118.12	1297	815	-	-	-
	70	117	118.72	13791	97	118.63	137	356	-	-	-
150	5	124	135.11	-	-	128.74	31077	2954	-	-	-
	10	136	142.81	-	-	139.59	6089	3247	-	-	-
	20	141	146.81	-	-	145.12	173425	61639	-	-	-
	30	144	147.38	-	-	146.67	3043	2617	-	-	-
	50	146	148.38	2104992	2190	148.10	1755	2756	-	-	-
	70	147	148.72	21625	301	148.63	219	1828	-	-	-
200	50	196	198.32	-	-	198.07	3125	20155	-	-	-
	70	197	198.73	38215	1322	198.63	253	8154	-	-	-

UNICAMP

Computational Results

$\|V\|$	dens.	opt.	B&B LP bound	B&B # nodes	B&B T(s)	Dir LP bound	Dir # nodes	Dir T(s)	SAP LP bound	SAP # nodes	SAP T(s)
100	5	76	83.42	-	-	79.36	605	24.5	78.57	-	-
	10	87	93.27	-	-	89.52	135	9.36	-	-	-
	20	92	96.56	-	-	94.85	1025	86.1	-	-	-
	30	94	97.39	-	-	96.68	1753	258	-	-	-
	50	96	98.36	348389	213	98.03	479	132	-	-	-
	70	97	98.76	9091	50.5	98.64	121	154	-	-	-
120	5	95	105.04	-	-	97.77	24	2.65	-	-	-
	10	107	113.16	-	-	109.83	869	65.4	-	-	-
	20	112	116.39	-	-	114.93	2401	393	-	-	-
	30	114	117.40	-	-	116.69	2301	653	-	-	-
	50	116	118.42	571335	435	118.12	1297	815	-	-	-
	70	117	118.72	13791	97	118.63	137	356	-	-	-
150	5	124	135.11	-	-	128.74	31077	2954	-	-	-
	10	136	142.81	-	-	139.59	6089	3247	-	-	-
	20	141	146.81	-	-	145.12	173425	61639	-	-	-
	30	144	147.38	-	-	146.67	3043	2617	-	-	-
	50	146	148.38	2104992	2190	148.10	1755	2756	-	-	-
	70	147	148.72	21625	301	148.63	219	1828	-	-	-
200	50	196	198.32	-	-	198.07	3125	20155	-	-	-
	70	197	198.73	38215	1322	198.63	253	8154	-	-	-

UNICAMP

Conclusions

Fujie's algorithm:

- Weaker upper (LP) bounds.
- Demanding in terms of memory.
- Faster for some high density instances.

Directed graph reformulation:

- Stronger LP bounds than original undirected formulation.
- Faster in 31 out of 37 instances tested.
- Attempted to solve larger number of test instances.

SAP reformulation:

- Stronger LP bounds than other formulations.
- Highly demanding in terms of CPU time.

Conclusions

Fujie's algorithm:

- Weaker upper (LP) bounds.
- Demanding in terms of memory.
- Faster for some high density instances.

Directed graph reformulation:

- Stronger LP bounds than original undirected formulation.
- Faster in 31 out of 37 instances tested.
- Attempted to solve larger number of test instances.

SAP reformulation:

- Stronger LP bounds than other formulations.
- Highly demanding in terms of CPU time.

Conclusions

Fujie's algorithm:

- Weaker upper (LP) bounds.
- Demanding in terms of memory.
- Faster for some high density instances.

Directed graph reformulation:

- Stronger LP bounds than original undirected formulation.
- Faster in 31 out of 37 instances tested.
- Attempted to solve larger number of test instances.

SAP reformulation:

- Stronger LP bounds than other formulations.
- Highly demanding in terms of CPU time.

Conclusions

Fujie's algorithm:

- Weaker upper (LP) bounds.
- Demanding in terms of memory.
- Faster for some high density instances.

Directed graph reformulation:

- Stronger LP bounds than original undirected formulation.
- Faster in 31 out of 37 instances tested.
- Attempted to solve larger number of test instances.

SAP reformulation:

- Stronger LP bounds than other formulations.
- Highly demanding in terms of CPU time.

UNICAMP

Conclusions

Fujie's algorithm:

- Weaker upper (LP) bounds.
- Demanding in terms of memory.
- Faster for some high density instances.

Directed graph reformulation:

- Stronger LP bounds than original undirected formulation.
- Faster in 31 out of 37 instances tested.
- Attempted to solve larger number of test instances.

SAP reformulation:

- Stronger LP bounds than other formulations.
- Highly demanding in terms of CPU time.

UNICAMP

Conclusions

Fujie's algorithm:

- Weaker upper (LP) bounds.
- Demanding in terms of memory.
- Faster for some high density instances.

Directed graph reformulation:

- Stronger LP bounds than original undirected formulation.
- Faster in 31 out of 37 instances tested.
- Attempted to solve larger number of test instances.

SAP reformulation:

- Stronger LP bounds than other formulations.
- Highly demanding in terms of CPU time.

Conclusions

Fujie's algorithm:

- Weaker upper (LP) bounds.
- Demanding in terms of memory.
- Faster for some high density instances.

Directed graph reformulation:

- Stronger LP bounds than original undirected formulation.
- Faster in 31 out of 37 instances tested.
- Attempted to solve larger number of test instances.

SAP reformulation:

- Stronger LP bounds than other formulations.
- Highly demanding in terms of CPU time.

Conclusions

Fujie's algorithm:

- Weaker upper (LP) bounds.
- Demanding in terms of memory.
- Faster for some high density instances.

Directed graph reformulation:

- Stronger LP bounds than original undirected formulation.
- Faster in 31 out of 37 instances tested.
- Attempted to solve larger number of test instances.

SAP reformulation:

- Stronger LP bounds than other formulations.
- Highly demanding in terms of CPU time.